Homework 1: Reproducible Computing

BIOS 731 – Advanced Statistical Computing

Context and learning objectives

This assignment reinforces the core ideas from **Module 1: Reproducible Computing in R**. The goal is not merely to write correct R code, but to demonstrate that you can organize, document, and version-control a computational analysis so that it can be **reproduced by someone else from scratch**.

By completing this assignment, you should be able to:

- Create a well-structured R project that separates raw data, code, and outputs.
- Use Git and GitHub to track and communicate the evolution of an analysis.
- Write a fully reproducible report using R Markdown or Quarto.
- Use relative paths and project-based workflows rather than ad-hoc file locations.
- Communicate the structure and purpose of a data analysis project via a README.

Due Date and Submission

Submit one link on Canvas: the public GitHub repository that contains your project for this assignment.Your grade will be based on what is present in that repository at the due date. Any Git commits with timestamps after the due date will be considered late.

In addition, your repository must contain a knitted **PDF report** generated from a `.Rmd` or `.qmd` file containing your report for the project.

Important constraints:

- Your R Markdown or Quarto document must not have calls to `install.packages` within the document- just load the packages in an obvious place and document which ones you will be using.
- It should only load the packages necessary for your analysis.
- All file access must use relative paths within the project directory.

Point distribution

Component	Points
Project setup	10
Data cleaning	10
Modeling	10
Visualization	10
README & documentation	20
Reproducibility	40

Problem 1 — Reproducible project

You are being graded on the entire structure of your submission and how well you apply the principles of reproducible computing using:

- R Projects
- Git and GitHub
- R Markdown or Quarto
- A clean and logical file organization
- Self-documenting code

You will create a small but complete data analysis project.

1.1 Project setup

- Create a public GitHub repository and a corresponding local R Project. Use the naming convention `bios731_hw1_<YourLastName>`. For example, mine would be `bios731_hw1_wrobel`
- Use Git to track your work. You should have multiple commits that reflect a reasonable workflow (e.g., adding data, writing cleaning code, fitting models, polishing the report).
- Push the entire project to GitHub.

1.2 Dataset and analysis

Choose one dataset to analyze. This may come from your own research or from a public source, but it must be legally and ethically shareable in a public GitHub repository.

Your project must include the following components:

- **Data reading and cleaning**

- Place the raw data in a `data/` directory.
- Write `R` code that:
    * Reads the data using relative paths
    * Performs any necessary data cleaning
    * Produces and saves a tidy dataset for analysis

- **Modeling**

    - Fit at least one statistical model that is appropriate for your data and research question. Doesn't need to be anything fancy- linear regression is fine.

- **Visualization**

    - Produce at least one meaningful visualization that supports or explains your analysis.

- **Final Report**

    - Include a final report that is a R Markdown or Quarto document that should:
        * Describe the dataset and research question
        * Show and explain the cleaning steps
        * Present the model(s)
        * Include the visualization(s)
        * Summarize the findings

1.3 README

Your GitHub repository must contain a `README.md` file that explains:

- What the dataset is and where it comes from
- What each major file and folder in the project does
- How someone else can reproduce your analysis
- The output of `sessionInfo()`, including what libraries and library versions are used in your analysis.

1.4 Reproducibility criteria

To get full credit for this assignment, we must be able to knit and obtain the same final report without changing file paths, installing packages inside the document, or manually fixing errors.